REMARKS

Applicant respectfully requests reconsideration and allowance of the subject application. Claims 1-40 are pending in this application.

35 U.S.C. § 112

Claim 37 stands rejected under 35 U.S.C. §112, second paragraph. Applicant respectfully disagrees.

The specification states, at page 7, lines 11-15, that:

> However, the invention is not limited to use in implementing access control or to use within an operating system. The improved hashing structure described herein can be used in any of a wide variety of locations, including many of those where conventional hashing structures currently exist.

The claims, as originally filed, list a resource manager that is not part of an operating system as an example of such a use that is not within an operating system (see, page 29, lines 1-2).

Thus, for at least these reasons, Applicant respectfully submits that claim 37 complies with 35 U.S.C. §112, second paragraph.

Applicant respectfully requests that the §112 rejections be withdrawn.

35 U.S.C. § 103

Claims 1-7 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,505,191 to Baclawski (hereinafter "Baclawski") in view of U.S. Patent No. 5,852,821 to Chen et al. (hereinafter "Chen") and further in view of Applicant's Admitted Prior Art (hereinafter "AAPA"). Applicant respectfully submits that claims 1-7 are not obvious over Baclawski in view of Chen and AAPA.

Baclawski is directed to a distributed computer database system and method employing hypertext linkage analysis (see, Title). In Baclawski, a user transmits a query from a user computer to a front end computer (see, col. 7, lines 26-29). After verifying that the query is acceptable, the front end computer performs any reformatting necessary to make the query compatible with the requirements of the search engine, and transmits the query to one of the home nodes of the search engine (see, col. 7, lines 37-41). The home node parses the query into a series of elementary queries (see, col. 7, lines 44-45). Elementary queries are of three kinds: index queries, link queries, and object queries (see, col. 7, lines 45-46). The home node extracts information from each elementary query depending on what kind of elementary query it is (see, col. 7, line 66 – col. 8, line 1). The home node extracts the OID (object identifier) from an object query, and extracts features from an index query or a link query (see, col. 8, lines 1-3, and col. 2, line 17). Features are extracted from structured elementary queries or documents by parsing the document to produce a data structure, then dividing this data structure into (possibly overlapping) substructures called fragments (see, col. 8, lines 4-7). The home node encodes each fragment of the query by using a predefined hashing function (see, col. 8, lines 61-62). Data in the distributed computer database system was previously stored locally on the various index nodes using this hashing function to generate an index to the data in the local database (see, col. 8, lines 62-66). Index nodes whose hashed feature fragments match the index feature fragments by which the data was initially stored on that index node respond to the elementary query by transmitting the OIDs matching the index terms of the requested information to the home node (see, col. 9, lines 26-30).

Chen is directed to a high-speed data base query method and apparatus (see, Title). In Chen, an index of bit vectors is created by accessing one of the values stored in a database and assigning each bit of the bit pattern for that value, from the most significant bit to the least significant bit, to a unique position in successive bit vectors (see, col. 4, lines 31-39). This accessing and assigning is repeated for each remaining value to form an index of bit vectors for the values (see, col. 4, lines 39-42). In order to search the index thus created for retrieving and/or reconstructing those data values greater than a search value, these bit vectors are used, in conjunction with the search value, to generate multiple answer vectors (see, col. 5, line 31- col. 6, line 13).

Applicant respectfully submits that the rejection of claim 1 under U.S.C. §103(a) fails to establish a *prima facie* case of obviousness, and therefore should be withdrawn. As set forth in MPEP §§2142 and 2143, a *prima facie* case of obviousness has three basic requirements, including: a) there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings; b) there must be a reasonable expectation of success; and c) the prior art references must teach all of the claim limitations. The Applicant respectfully submits that the proposed combination of Baclawski, Chen, and AAPA fails to meet at least one of requirements a) through c), as they pertain to pending claim 1, and therefore no *prima facie* case of obviousness has been established.

Applicant respectfully submits that at least requirements a) and c) for establishing a *prima facie* case of obviousness have not been met. That is, Applicant respectfully submits that there is no suggestion or motivation provided by any of the references or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings, and further submits that even if combined, the references do not teach all features of the claims.

In the April 22 Office Action, at p. 3, it was asserted that:

> As for generating a hash key based on the input value and separating the hash key into a plurality of portions, Baclawski discloses an alternative means wherein the input value is separated into fragments and each of the fragments are hashed (see for example; col. 8 ln 4-9 and ln 61-67). Baclawski uses this method as a means of indexing into a plurality of sub-hashes (data in local database). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to have a design choice in generating a hash key on the input value and separating the hash key into a plurality of portions because the applicant has not explicitly stated any other reason or purpose for such means other than for generating an index to lookup values and the means disclosed by Baclawski is just as efficient.

Applicant respectfully disagrees with this assertion and respectfully submits that there is no suggestion or motivation to modify Baclawski as asserted in the April 22 Office Action.

As set forth in MPEP §2141, references must be viewed without the benefit of impermissible hindsight vision afforded by the Applicant's disclosure. However, it appears from the April 22 Office Action that an improper hindsight reconstruction is being relied on in the rejection of claim 1. The April 22 Office Action states that "It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention . . . because the applicant has not explicitly stated any other reason or purpose for such means . . . ." From this language, it appears that Applicant's disclosure is being used as a basis for modifying Baclawski. Such reliance on Applicant's disclosure is improper, as any suggestion or motivation to modify Baclawski must be found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, not in Applicant's disclosure.

Additionally, assuming for the sake of argument that the references were combined, Applicant respectfully submits that the combination does not disclose or suggest claim 1. Claim 1 recites in part:

> generating a hash key based on the input value;
> separating the hash key into a plurality of portions;
> indexing into each of a plurality of sub-hashes using one of the plurality of portions;

In the April 22 Office Action, at p. 3, Baclawski is relied on as disclosing the generating, separating, and indexing of claim 1. Applicant respectfully disagrees with this assertion, and respectfully submits that there is no disclosure or suggestion in Baclawski of the generating, separating, and indexing as recited in claim 1.

As discussed above, Baclawski describes parsing a query from a user into multiple elementary queries, extracting features from a query, and dividing the data structure produced from extracting the features into fragments. However, Applicant respectfully submits that none of this parsing, extracting, or dividing discloses or suggests separating a hash key into a plurality of portions as recited in claim 1. The mere disclosure of parsing, extracting, or dividing some other type of data does not disclose or suggest separating a hash key into a plurality of portions as recited in claim 1. Absent some mention or suggestion to separate a hash key into a plurality of portions, Applicant respectfully submits that Baclawski cannot disclose or suggest separating a hash key into a plurality of portions as recited in claim 1.

With respect to Chen, it appears that Chen is being relied on as disclosing combining the plurality of values to generate a hash result, wherein each bit in the hash result corresponds to one of the plurality of target values as recited in claim 1. Applicant respectfully disagrees with this characterization of Chen.

Chen does not discuss hashing or hash results, much less combining a plurality of values to generate a hash result. The word "hash" is not even present in Chen. Absent any discussion or even mention of hashing or a hash result, Applicant respectfully submits that Chen cannot disclose or suggest combining the plurality of values to generate a hash result, wherein each bit in the hash result corresponds to one of the plurality of target values as recited in claim 1.

Furthermore, the plurality of values that the combining refers to are the plurality of values identified from the plurality of sub-hashes based on the indexing as recited in claim 1. As there is no discussion or even mention of hashing in Chen, Applicant respectfully submits that Chen cannot disclose or suggest combining a plurality of values identified from the plurality of sub-hashes based on the indexing as recited in claim 1.

With respect to AAPA, AAPA is not cited as curing, and does not cure, the deficiencies of Baclawski and Chen as discussed above.

For at least these reasons, Applicant respectfully submits that claim 1 is allowable over Baclawski in view of Chen and AAPA.

Given that claims 2-5 and 7 depend from claim 1, Applicant respectfully submits that claims 2-5 and 7 are likewise allowable over Baclawski in view of Chen and AAPA for at least the reasons discussed above with respect to claim 1.

With respect to claim 6, claim 6 depends from claim 1 and Applicant respectfully submits that claim 6 is allowable over Baclawski in view of Chen and AAPA for at least the reasons discussed above with respect to claim 1. Furthermore, claim 6 recites:

> One or more computer readable media as recited in claim 1, wherein the separating comprises separating the hash key into a plurality of contiguous and equal portions.

Applicant respectfully submits that Baclawski in view of Chen and AAPA does not disclose or suggest separating comprises separating the hash key into a plurality of contiguous and equal portions as recited in claim 6.

In the April 22 Office Action at p. 7, Baclawski at col. 8 lines 49-60 is cited as disclosing the separating of claim 6. Applicant respectfully disagrees with this characterization of Baclawski. The cited portion of Baclawski states that the possible value of a fragment (such as the brightness of an image) will be partitioned into a collection of contiguous, non-overlapping ranges of values (see, col. 8, lines 48-51). The cited portion of Baclawski is discussing partitioning a fragment, not separating a hash key as recited in claim 6. Furthermore, there is no mention in this partitioning of the fragment being partitioned into contiguous and equal portions, and thus there cannot be any disclosure or suggestion of separating a hash key into a plurality of contiguous and equal portions as recited in claim 6.

With respect to Chen and AAPA, Chcn and AAPA are not cited as curing, and do not cure, the deficiencies of Baclawski as discussed above.

For at least these reasons, Applicant respectfully submits that claim 6 is allowable over Baclawski in view of Chen and AAPA.

Claims 8-18 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Chen in view of U.S. Patent No. 5,852,822 to Srinivasan et al. (hereinafter "Srinivasan"). Applicant respectfully submits that claims 8-18 are not obvious over Chen in view of Srinivasan.

Applicant respectfully submits that the rejection of claim 8 under U.S.C. §103(a) fails to establish a *prima facie* case of obviousness, and therefore should be withdrawn. Applicant respectfully submits that there is no suggestion or motivation provided by any of the references or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings, and further submits that even if combined, the references do not teach all features of the claims.

In the April 22 Office Action at p. 9, it was asserted that:

> It would have been obvious to one of ordinary skill in the art of at the time of the applicant's invention to combine the teachings of Srinivasan within the teachings of Chen because it would have speed up the processing and lookup of sub-hashes (bit vectors) in the index and thus increase processing efficiency and speed.

Applicant respectfully disagrees with this assertion.

Chen is directed to a particular technique for creating an index of bit vectors, which are shown as 44a-44d in Fig. 4 of Chen, and assigning values to those bit vectors (see, col. 4, lines 31-63). These bit vectors can then be searched for retrieving and/or reconstructing data values (see, col. 5, lines 31-34).

Srinivasan, on the other hand is directed to a method and apparatus for building, maintaining, and using a multi-level index (see, col. col. 2, lines 8-39, and col. 11, line 43 – col. 12, line 21). Chen and Srinivasan are thus directed to two very different techniques for creating indexes. As discussed in Chen, the bit vectors are searched in a particular manner for retrieving and/or reconstructing data values (see, col. 5, line 31 – col. 6, line 13). There is no discussion of how the index of bit vectors of Chen could be modified to incorporate the multi-level index of Srinivasan and still allow for this particular manner of searching. As such, Applicant respectfully submits that it would not have been obvious to one of ordinary skill in the art to combine Chen and Srinivasan.

Additionally, assuming for the sake of argument that the references were combined, Applicant respectfully submits that the combination does not disclose or suggest claim 8. Claim 8 recites:

> A hashing architecture comprising:
> a plurality of sub-hashes;
> a plurality of sub-hash indexes, each index being generated from a hash key and used to index into one of the plurality of sub-hashes; and
> a combiner coupled to receive values from the plurality of sub-hashes based on the plurality of sub-hash indexes, and to generate a hash result by combining the received values.

Applicant respectfully submits that Chen in view of Srinivasan does not disclose or suggest a hashing architecture as recited in claim 8.

In the April 22 Office Action, at p. 8, Chen is recited as teaching all of the elements of claim 8 except for the "index begin generated from a hash key". Applicant respectfully disagrees with this characterization of Chen. Chen discusses creating a plurality of bit vectors 44a-44d (see, col. 4, lines 45-46). The number of bit vectors created equals the length of the bit patterns for the values, so if the memory allocates 32 bits per character (or digit) for each value, then 32 bit vectors are created (see, col. 4, lines 46-50). Each value in the memory is represented by a 32-bit bit pattern, so the number 3 has a bit pattern 0…011 (see, col. 4, lines 53-56). For the first value, each bit of the bit pattern from the most significant bit to the least significant bit is assigned by the server to the first position in each of the bit vectors 44a-44d, so the most significant bit for the number 3 is assigned to the first position of the first vector 44d and the least significant bit is assigned to the first position of the last bit vector 44a (see, col. 4, lines 56-62). This is illustrated in Fig. 4 of Chen, where the bit pattern 0…011 for the value 3 is spread across the first positions in the bit vectors 44a-44d.

From the April 22 Office Action at p. 8, it appears that these bit vectors 44a-44d are being relied on as disclosing the plurality of sub-hashes of claim 8. If these bit vectors were to be the plurality of sub-hashes of claim 8 as asserted in the April 22 Office Action, then following the language of claim 8 there would need to be some plurality of sub-hash indexes used to index into one of the bit vectors disclosed in Chen. However, there is no discussion or suggestion in Chen of any such plurality of sub-hash indexes. As stated in the cited portion of Chen, the processing includes "repeating the above-described accessing and assigning steps for each remaining value of the set to form an index of bit vectors for the values (steps 38, 40)" (see, col. 4, lines 38-41). These bit vectors form an index 46 which can be searched (see, Fig. 4 and col. 5, lines 31-34). Thus, the bit vectors themselves form the index in Chen; there is not a separate index into the bit vectors described or mentioned in Chen. As such, Applicant respectfully submits that Chen cannot disclose or suggest a plurality of sub-hash indexes used to index into one of the plurality of sub-hashes as recited in claim 8.

Furthermore, as there is no plurality of sub-hash indexes disclosed or suggested in Chen, Applicant respectfully submits that Chen cannot disclose or suggest a combiner coupled to receive values from the plurality of sub-hashes based on the plurality of sub-hash indexes as recited in claim 8.

With respect to Srinivasan, Srinivasan is not cited as curing, and does not cure, these deficiencies of Chen. As such, Applicant respectfully submits that claim 8 is allowable over Chen in view of Srinivasan for at least these reasons.

Given that claims 9-13 depend from claim 8, Applicant respectfully submits that claims 9-13 are likewise allowable over Chen in view of Srinivasan for at least the reasons discussed above with respect to claim 8.

With respect to claim 14, Applicant respectfully submits that, similar to the discussion above regarding claim 8, it would not have been obvious to combine Chen and Srinivasan. Furthermore, even if Chen and Srinivasan were combined, Applicant respectfully submits that, similar to the discussion above regarding claim 8, the combination does not disclose or suggest identifying a plurality of values from a plurality of sub-hashes by indexing into each of the plurality of sub-hashes using one of the plurality of sub-hash keys, and generating a hash result based on the plurality of values as recited in claim 14.

For at least these reasons, Applicant respectfully submits that claim 14 is allowable over Chen in view of Srinivasan.

Given that claims 15-18 depend from claim 14, Applicant respectfully submits that claims 15-18 are likewise allowable over Chen in view of Srinivasan for at least the reasons discussed above with respect to claim 14.

Claims 19-37 stand rejected under 35 U.S.C. §103(a) as being unpatentable over AAPA in view of Chen and further in view of Srinivasan. Applicant respectfully submits that claims 19-37 are not obvious over AAPA in view of Chen and further in view of Srinivasan.

With respect to claim 19, Applicant respectfully submits that, similar to the discussion above regarding claim 8, it would not have been obvious to combine Chen and Srinivasan. Furthermore, even if Chen and Srinivasan were combined, Applicant respectfully submits that, similar to the discussion above regarding claim 8, the combination does not disclose or suggest indexing into a first sub-hash using the first portion to identify a first sub-hash value, indexing into a second sub-hash using the second portion to identify a second sub-hash value, and combining the first sub-hash value and the second sub-hash value to generate a result value as recited in claim 19. In addition, Applicant respectfully submits that AAPA is not cited as curing, and does not cure, these deficiencies of Chen in view of Srinivasan.

For at least these reasons, Applicant respectfully submits that claim 19 is allowable over AAPA in view of Chen and further in view of Srinivasan.

Given that claims 20-22 depend from claim 19, Applicant respectfully submits that claims 20-22 are likewise allowable over AAPA in view of Chen and further in view of Srinivasan for at least the reasons discussed above with respect to claim 19.

With respect to claim 23, Applicant respectfully submits that, similar to the discussion above regarding claim 8, it would not have been obvious to combine Chen and Srinivasan. Furthermore, even if Chen and Srinivasan were combined, Applicant respectfully submits that, similar to the discussion above regarding claim 8, the combination does not disclose or suggest indexing into each of a plurality of sub-hashes using a respective one of the plurality of sub-hash indexes, and generating a result hash value by combining the plurality of values resulting from indexing into the plurality of sub-hashes as recited in claim 23. In addition, Applicant respectfully submits that AAPA is not cited as curing, and does not cure, these deficiencies of Chen in view of Srinivasan.

For at least these reasons, Applicant respectfully submits that claim 23 is allowable over AAPA in view of Chen and further in view of Srinivasan.

Given that claims 24-30 depend from claim 23, Applicant respectfully submits that claims 24-30 are likewise allowable over AAPA in view of Chen and further in view of Srinivasan for at least the reasons discussed above with respect to claim 23.

With respect to claim 31, Applicant respectfully submits that, similar to the discussion above regarding claim 8, it would not have been obvious to combine Chen and Srinivasan. Furthermore, even if Chen and Srinivasan were combined, Applicant respectfully submits that, similar to the discussion above regarding claim 8, the combination does not disclose or suggest indexing into each of the plurality of sub-hashes using a respective one of the plurality of sub-hash indexes, and combining the plurality of values to generate a hash result value, wherein each bit in the hash result value corresponds to one of the plurality of security token security identifiers as recited in claim 31. In addition, Applicant respectfully submits that AAPA is not cited as curing, and does not cure, these deficiencies of Chen in view of Srinivasan.

For at least these reasons, Applicant respectfully submits that claim 31 is allowable over AAPA in view of Chen and further in view of Srinivasan.

Given that claims 32-37 depend from claim 31, Applicant respectfully submits that claims 32-37 are likewise allowable over AAPA in view of Chen and further in view of Srinivasan for at least the reasons discussed above with respect to claim 31.

Claims 38-40 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Chen. Applicant respectfully submits that claims 38-40 are not obvious over Chen.

Claim 38 recites:

> A method comprising:
> for each sub-hash in a plurality of sub-hashes that can be used together to generate a hash result,
> (a) identifying a bit in a location of a sub-hash,
> (b) identifying, in a source value, a plurality of bits corresponding to the sub-hash,
> (c) comparing an identifier of the location to the plurality of bits,
> (d) setting the bit if the identifier of the location matches the plurality of bits, and otherwise clearing the bit, and
> (e) repeating acts (a), (b), (c), and (d) for each of a plurality of bits in the location of the sub-hash.

In the April 22 Office Action at p. 25, Chen at col. 4 lines 30-64 is cited as disclosing the acts (a), (b), (c), (d), and (e) of claim 38. Applicant respectfully disagrees with this characterization of Chen.

The cited portion of Chen discusses creating a plurality of bit vectors 44a-44d (see, col. 4, lines 45-46). The number of bit vectors created equals the length of the bit patterns for the values, so if the memory allocates 32 bits per character (or digit) for each value, then 32 bit vectors are created (see, col. 4, lines 46-50). Each value in the memory is represented by a 32-bit bit pattern, so the number 3 has a bit pattern 0...011 (see, col. 4, lines 53-56). For the first value, each bit of the bit pattern from the most significant bit to the least significant bit is assigned by the server to the first position in each of the bit vectors 44a-44d, so the most significant bit for the number 3 is assigned to the first position of the first vector 44d and the least significant bit is assigned to the first position of the last bit vector 44a (see, col. 4, lines 56-62). This is illustrated in Fig. 4 of Chen, where the bit pattern 0...011 for the value 3 is spread across the first positions in the bit vectors 44a-44d.

From the April 22 Office Action at p. 25, it appears that these bit vectors 44a-44d are being relied on as disclosing the sub-hash of claim 38. However, referring to these bit vectors of Chen, there is no comparing an identifier of the location to the plurality of bits and setting the bit if the identifier of the location matches the plurality of bits, and otherwise clearing the bit. As discussed above, the bit pattern for a particular value in Chen is spread across a particular position in the bit vectors of Chen. There is no comparing of an identifier of a location to a plurality of bits and setting a bit based on whether the location matches the plurality of bits as recited in claim 38. Rather, in Chen it is simply the bit value for thc value being spread across the bit vectors. For example, as shown in Fig. 4 of Chen, the value of 3 is assigned to the bit vectors with the "11", which represents the value 3, being stored in the two vectors 44a and 44b, with all other bit vectors 44c and 44d storing "0". There is not any comparing to determine what value to store in the bit vectors of Chen – a "0" is assigned to a particular position in a particular bit vector if the corresponding position in the bit pattern is a "0", and a "1" is assigned to a particular position in a particular bit vector if the corresponding position in the bit pattern is a "1".

For at least these reasons, Applicant respectfully submits that claim 38 is allowable over Chen.

Given that claims 39-40 depend from claim 38, Applicant respectfully submits that claims 39-40 are likewise allowable over Chen for at least the reasons discussed above with respect to claim 38.

Applicant respectfully requests that the §103 rejections be withdrawn.

Conclusion

Claims 1-40 are in condition for allowance. Applicant respectfully requests reconsideration and issuance of the subject application. Should any matter in this case remain unresolved, the undersigned attorney respectfully requests a telephone conference with the Examiner to resolve any such outstanding matter.

Respectfully Submitted,

Date: 6/22/04

By:
Allan T. Sponseller
Reg. No. 38,318
(509) 324-9256